

## Homework Exercises 4

Dr. Sarvar Abdullaev  
s.abdullaev@inha.uz

October 4, 2022

After completing all lab exercises below, zip all your solution files into a single archive with your student ID (e.g. u210023.zip) and upload it to eClass before the deadline.

Note, you do not have to submit other lab materials such as lab slides or examples.

### 1 Experiments with Brookshear Machine

Follow the instructions given here for installation of Brookshear Machine on your computer. Note, you must install Java first, and then download following JAR file to your computer. You should see Figure 1 on your screen after running this JAR file.

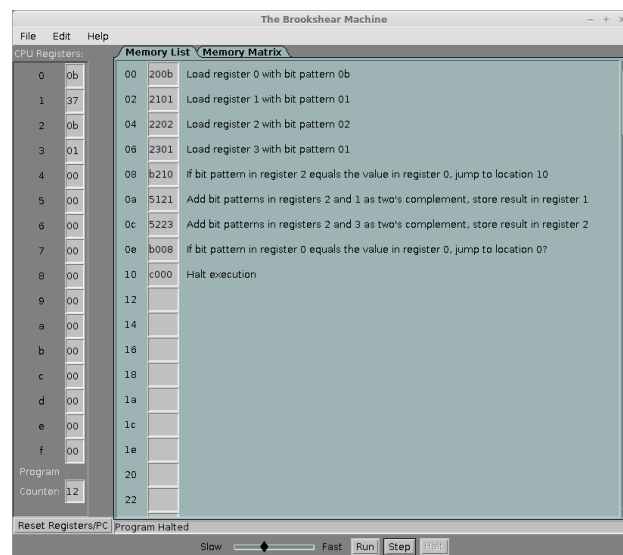


Figure 1: Brookshear Machine

Input following programs into your Brookshear Machine and evaluate their execution.

#### 1.1 Program 1

Suppose the memory cells at addresses 00 through 09 in the Brookshear Machine contain the following bit patterns:

Address	Contents
00	1C
01	03
02	2B
03	03
04	5A
05	BC
06	3A
07	00
08	C0
09	00

Assume that the machine starts with its program counter containing 00.

1. What will be in the memory cell at address 00 when the machine halts?
2. What bit pattern will be in the program counter when the machine halts?

Save your results to `ps1-1.txt`.

## 1.2 Program 2

Suppose the memory cells at addresses 00 through 0D in the machine contain the following bit patterns:

Address	Contents
00	20
01	04
02	21
03	01
04	40
05	12
06	51
07	12
08	B1
09	0C
0A	B0
0B	06
0C	C0
0D	00

Assume that the machine starts with its program counter containing 00.

1. What bit pattern will be in register 0 when the machine halts?
2. What bit pattern will be in register 1 when the machine halts?
3. What bit pattern is in the program counter when the machine halts?

Save your results to `ps1-2.txt`.

## 1.3 Program 3

Suppose the memory cells at addresses F0 through FD in the machine contain the following (hexadecimal) bit patterns:

Address	Contents
F0	20
F1	00
F2	22
F3	02
F4	23
F5	04
F6	B3
F7	FC
F8	50
F9	02
FA	B0
FB	F6
FC	C0
FD	00

If we start the machine with its program counter containing F0, what is the value in register 0 when the machine finally executes the halt instruction at location FC?

Save your results to **ps1-3.txt**.

## 1.4 Program 4

In each of the following cases, write a short program in the machine language to perform the requested activities. Assume that each of your programs is placed in memory starting at address 00.

1. Move the value at memory location D8 to memory location B3. Save your program as **ps1-4a.txt**.
2. Interchange the values stored at memory locations D8 and B3. Save your program as **ps1-4b.txt**.
3. If the value stored in memory location 44 is 00, then place the value 01 in memory location 46; otherwise, put the value FF in memory location 46. Save your program as **ps1-4c.txt**.

## 2 Python Tasks

Watch **Week 3: Simple Algorithms** lecture videos on this course. Then complete following Python tasks.

1. Run a Python program below to compute check if a given word is palindrome. Save your program as **ps2-1.py**.

```
word = input('Enter your word? ')
isPalindrome = True

# half length of the word
hl = int(len(word) / 2)

for i in range(0, hl):
    if word[i].lower() != word[-(i + 1)].lower():
        isPalindrome = False
        break

print('Is palindrome: ', isPalindrome)
```

2. Write a Python program which checks whether lowercase letters exist in a string. Save your program as **ps2-2.py**.
3. Write a Python program which counts the number of occurrences for each character in a string. Hint, use dict data structure in Python. Save your program as **ps2-3.py**.

4. Write a Python program which encodes any given string with alphabetic charecters into run-length encoded string. For example, if you provide a string `AAAAAABBBBCCCCCCCCDDDD`, it should return `7A4B8C3D`. Save your program as `ps2-4.py`.
5. Write a Python program which decodes any RLE string with alphabetic charecters into original format. For example, if you provide a string `7A4B8C3D`, it should return `AAAAAABBBBCCCCCCCCDDDD`. Save your program as `ps2-5.py`.